

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

1 1. (Cancelled)

1 2. (Previously Presented) The system of claim 9, wherein the event control module
2 is adapted to define a queue for a first one of the event objects, the queue having plural entries
3 corresponding to plural execution entities waiting on the event represented by the first event
4 object.

1 3. (Previously Presented) The system of claim 2, wherein the event control module
2 is adapted to further create second objects, wherein each entry of the queue comprises a link to a
3 corresponding second object, each execution entity to sleep on an associated second object to
4 wait on the event represented by the first event object.

1 4. (Original) The system of claim 3, wherein each second object is defined by a
2 condition variable.

1 5. (Original) The system of claim 4, wherein the controller signals each thread by
2 signaling the condition variable.

1 6. (Original) The system of claim 3, wherein each second object is defined by a
2 condition variable and a mutex.

1 7. – 8. (Cancelled)

1 9. (Currently Amended) A system comprising:
2 a Unix operating system;
3 a plurality of execution entities, the plurality of execution entities including a first
4 execution entity;
5 an event control module adapted to create event objects representing respective
6 events each having a state, the first execution entity to wait on plural events;
7 a data structure associated with the first execution entity, the data structure
8 containing information of the plural events that the first execution entity is waiting on, the data
9 structure further containing an indicator settable to one of plural values to specify respective
10 plural logical relationships between the plural events; and
11 a controller adapted to awaken the first execution entity by signaling the first
12 execution entity in response to one or more event state changes of the states of the plural events
13 according to the logical relationship specified by the indicator,
14 wherein each event object contains an indication of the state of the event, wherein
15 the indication has a first state to indicate that the corresponding event has been signaled and a
16 second state to indicate that the corresponding event has not been signaled,
17 wherein each event object has a type indication to indicate whether the event
18 object state indication is to be automatically reset to the second state from the first state once the
19 event has been signaled or to be manually reset to the second state from the first state by an
20 explicit action,
21 wherein if a particular one of the event objects is to be automatically reset, then
22 only one of plural execution entities waiting on the particular event object will be awakened in
23 response to the corresponding event being signaled, and
24 wherein if the particular event object is to be manually reset, then all execution
25 entities waiting on the particular event object will be awakened in response to the corresponding
26 event being signaled.

1 10. – 11. (Cancelled)

1 12. (Previously Presented) The system of claim 9, further comprising queues
2 associated with corresponding event objects representing events the first execution entity is
3 waiting on, each queue containing an entry corresponding to the first execution entity.

1 13. (Previously Presented) The system of claim 12, wherein the event control module
2 is adapted to define a barrier object, the first execution entity to sleep on the barrier object to
3 wait on the plural events, the queue of each event object containing a link to the barrier object.

1 14. (Original) The system of claim 13, wherein the barrier object is defined at least
2 by a condition variable.

1 15. (Original) The system of claim 13, wherein the barrier object is defined at least
2 by a condition variable and a mutex.

1 16. (Previously Presented) The system of claim 9, wherein the event control module
2 comprises a library.

1 17. (Previously Presented) The system of claim 9, wherein the execution entities
2 comprise threads.

1 18. (Original) The system of claim 17, further comprising plural processes, each
2 process associated with one or more threads,
3 the event control module to create a local event to synchronize threads within a
4 process and to create a global event to synchronize threads of different processes.

1 19. (Original) The system of claim 18, wherein the global event comprises a named
2 event.

1 20. (Previously Presented) The system of claim 9, further comprising a plurality of
2 nodes, each node comprising one or more of the plurality of execution entities.

1 21. (Previously Presented) An article comprising at least one storage medium
2 containing instructions for providing event-based synchronization in a system in which execution
3 entities are running, the instructions when executed causing the system to:
4 generate event objects in a Unix operating system environment representing
5 events used for synchronizing execution entities in the system, each event object having a state to
6 indicate if the corresponding event has been signaled;
7 provide a queue containing entries associated with a first event object, each entry
8 associated with a corresponding execution entity, the plural entries of the queue enabling plural
9 execution entities to wait on the first event object; and
10 selectively set a type variable to one of a first value and a second value, the first
11 value indicating that the first event object is of an auto-reset type, and the second value
12 indicating that the first event object is of a manual reset type;
13 in response to the state of the first event object indicating the corresponding event
14 has been signaled,
15 automatically clear the state of the first event object to an un-sigaled state
16 and awaken only one of the plural execution entities waiting on the first event object in response
17 to the type variable being set to the first value, and
18 not clear the state of the first event object until manually cleared and
19 awaken all threads waiting on the first event object in response to the type variable being set to
20 the second value.

1 22. (Original) The article of claim 21, wherein the instructions when executed cause
2 the system to further create barrier objects, each execution entity waiting on a corresponding
3 barrier object to wait on an event.

1 23. (Previously Presented) The article of claim 22, wherein the instructions when
2 executed cause the system to create barrier objects by defining each barrier object based on a
3 condition variable according to the Unix operating system.

1 24. (Previously Presented) The article of claim 22, wherein the instructions when
2 executed cause the system to create barrier objects by defining each barrier object based on a
3 condition variable and mutex according to the Unix operating system.

1 25. (Previously Presented) The article of claim 22, wherein the queue of the first
2 event object contains entries pointing to the barrier objects of the plural execution entities
3 waiting on the first event object.

1 26. (Previously Presented) The article of claim 25, wherein the instructions when
2 executed cause the system to provide a routine associated with each event object, the routine of
3 the first event object to traverse the queue of the first event object and to signal the barrier
4 objects pointed to by the entries in the queue of the first event object.

1 27. – 38. (Cancelled)

1 39. (Previously Presented) The system of claim 9, wherein the indicator is settable to
2 a first value to specify a logical AND relationship between the plural events, and in response to
3 the first value of the indicator, the controller to awaken the first execution entity in response to
4 all of the plural events waited on by the first execution entity being signaled.

1 40. (Previously Presented) The system of claim 39, wherein the indicator is settable
2 to a second value to specify a logical OR relationship between the plural events, and in response
3 to the second value of the indicator, the controller to awaken the first execution entity in
4 response to any of the plural events waited on by the first execution entity being signaled.

1 41. (Previously Presented) The article of claim 21, wherein a first one of the
2 execution entities waits on plural events represented by respective event objects, the instructions
3 when executed causing the system to:

4 provide a data structure containing information of the plural events waited upon
5 by the first execution entity, the data structure further containing an indicator settable to one of
6 plural values to specify respective plural logical relationships between the plural events waited
7 on by the first execution entity; and

8 awaken the first execution entity in response to states of the plural events waited
9 upon by the first execution entity according to the logical relationship specified by the indicator.

1 42. (Previously Presented) The article of claim 41, wherein the instructions when
2 executed cause the system to set the indicator to a value to indicate a logical AND relationship,
3 wherein awakening the first execution entity is in response to all of the plural events waited upon
4 by the first execution entity being signaled.

1 43. – 44. (Cancelled)